RATIONALITY & COMPETITION
CRC TRR 190

From Friends to Foes: National Identity and Collaboration in Diverse Teams

Nadzeya Laurentsyeva (LMU Munich)

Discussion Paper No. 226

December 18, 2019

Collaborative Research Center Transregio 190 | www.rationality-and-competition.de
Ludwig-Maximilians-Universität München | Humboldt-Universität zu Berlin
Spokesperson: Prof. Dr. Klaus M. Schmidt, University of Munich, 80539 Munich, Germany
+49 (89) 2180 3405 | info@rationality-and-competition.de

From Friends to Foes: National Identity and Collaboration in Diverse Teams [a]

Nadzeya Laurentsyeva[b]

December 15, 2019

This project studies collaboration in highly skilled, nationally diverse teams. An unexpected international political conflict makes national diversity more salient among existing and potential team members. I exploit this natural experiment to quantify the role of social, identity-driven, costs for performance and formation of diverse teams. Using microdata from GitHub, the world's largest hosting platform for software projects, I estimate the causal impacts of a political conflict that burst out between Russia and Ukraine in 2014. I find that the conflict strongly reduced online cooperation between Russian and Ukrainian programmers. The conflict lowered the likelihood that Ukrainian and Russian programmers work in the same team and led to the performance decline of existing joint projects. I provide evidence that the observed effects were not driven by economic considerations. Rather, the conflict activated national identities and shifted programmers' taste for teammates and projects. My results highlight the role of identity-driven concerns that can distort existing and prevent future collaborations, otherwise profitable from an economic perspective.
JEL classification: D22, D74, F23, F51, J71

Keywords: Teams, Diversity, Conflict, National identity, Open source

[a]I thank Michele Battisti, Florian Englmaier, Thomas Fackler, Yvonne Giesing, participants of COPE, EEA, SIOE, Ohlstadt conferences, as well as several research seminars in Munich for helpful comments and suggestions. Financial support from the Deutsche Forschungsgemeinschaft through Research Training Group 1928 and through CRC TRR 190 (project number 280092119) is gratefully acknowledged.
[b]LMU Munich. E-mail: nadzeya.laurentsyeva@econ.lmu.de

1 Introduction

In a globalised world, with lower communication costs and fewer obstacles to labour mobility, diverse teams have become common in many organisations. These teams exist because the benefits of pooling diverse individuals together outweigh higher communication and coordination costs. Yet, managing a diverse team is challenging. The team members might identify themselves as being part of different social groups, for instance, based on nationality or political preferences. What if, suddenly, these social groups find themselves on the opposite sides of the barricades, for example, due to a political or an ethnic conflict? Even if the conflict does not directly influence the team's environment, it can change identity prescriptions of the team members, spur stronger in-group/out-group feelings, lower trust, and thus reduce the team's productivity. Is there causal evidence for such effects in real-life teams? Do identity-related conflicts exert the economic impact by harming the performance of previously efficient teams and by preventing otherwise profitable future collaborations?

This project uses a novel and relevant setting to answer the above questions. I study the consequences of an exogenous political conflict for the performance and formation of nationally diverse teams on GitHub, the world's largest hosting service for software projects. In particular, I exploit the unexpected conflict between Russia and Ukraine following the annexation of Crimea in March 2014 and analyse the impact of this conflict on the online collaboration between Ukrainian and Russian programmers.

Using monthly microdata from Github, I first show that the conflict exerted a strong and persistent negative effect on the overall Ukrainian-Russian collaboration as measured by Ukrainian contributions to Russian projects and vice versa. The effect is symmetric on the extensive margin: after the conflict, there are fewer unique users contributing to projects from a now hostile country. However, on the intensive margine, Ukrainian programmers react stronger: conditional on collaborating with Russians, they contribute to fewer Russian projects. The empirical approach, compirising a triple difference specification and an event study, allows to control for time-varying activity of Ukrainian and Russian programmers, as well as for time-varying quantity and quality of Ukrainian and Russian projects. This ensures that the estimated effect is not driven by the fact that Ukrainian and/or Russian programmers reduce their activity on GitHub after 2014, nor that the (perceived) quality of projects deteriorates. In addition, the analysis excludes Donbass and Crimea regions, which could be directly affected by the conflict. I also show that the results cannot be rationalised by career expectations and higher business transaction costs.

Rather, the drop in collaboration concords with the identity-based explanation following Akerlof and Kranton (2000). The political conflict enforced national identification and, thus, could have changed programmers' taste for teammates and projects associated with an opposing identity. Intuitively, such identity effect should be larger for programmers with initially stronger national identity. Although the strength of the national identity cannot be directly observed, I can exploit ethnic heterogeneity within Ukraine to construct a proxy. I differentiate whether a Ukrainian programmer comes from a city with high or low share of ethnic Ukrainians (according to 2001 Census). My analysis shows that the above effects are driven solely by collaborations between Russians and Ukrainian users from cities with high share of ethnic Ukrainians.

I further show that following the conflict, Ukrainian users became significantly less likely to join Russian teams and thus to contribute saliently to Russian projects as team members. Yet, I do not observe a symmetric effect on Russian users, suggesting that the conflict affected Ukrainian programmers more strongly, likely, through raising their social image concerns. Lastly, I show that the decrease in contributions led to the performance decline of the affected projects, in particularly those owned by Russian users.

The GitHub setting provides several advantages for my analysis. First, GitHub was specifically designed as an online platform for collaboration. It offers powerful infrastructure that allows multiple users from different locations to coordinate their efforts while working on the same project. Therefore, GitHub represents an ideal laboratory to study team interactions and performance. Second, GitHub features an environment with very low information asymmetries: individual contributions of team members are directly observable by other GitHub users, thus, alleviating free-riding concerns in teams. From a researcher's perspective, this also limits the number of available interpretations for the observed results. Third, since its launch in 2007, GitHub has grown to include almost 40 million users. Many technology companies use GitHub for both open-source and commercial projects. Hence, a virtual team on GitHub should be representative of a real highly skilled team working on complex tasks. Fourth, prior to the 2014 conflict, both Ukrainian and Russian programmers were well represented at the platform. Because of a similar (or the same) language and technical backgrounds, Ukrainians and Russians had often worked together on various projects. After March 2014, there were no major interruptions in the access to the Internet in Ukraine or Russia nor could the introduced bilateral sanctions between the countries prevent programmers from working on the US-based online platform. For my analysis, therefore, I can treat this conflict as exogenous to the virtual working environment on GitHub. Finally, I work with a detailed dataset, generated from

GitHub databases. The dataset contains information about all users, projects, and related activities ever registered on the platform over 2012-15.

This project contributes to several strands of literature. First, it relates to the literature on team performance. Bandiera et al. (2005, 2009) and Mas and Moretti (2009) emphasised the importance of social preferences and provided evidence that socially-connected teams may be more resilient, for instance, to free-riding or to coordination problems. Lazear (1999) and Prat (2002) discussed the optimality conditions for team diversity. Together these studies focused on the importance of peer interaction and outlined the trade-off related to diverse or socially-distant teams: potential productivity gains against collaboration challenges. My project adds to this literature by providing empirical evidence for one of the risks that diverse teams face: external events can exacerbate social differences within teams and hence inhibit the performance. Second, the project also contributes to the theoretical and experimental literature on social identity and group performance (Charness et al. 2007; Chen et al. 2014; Chen and Li 2009). These works have established the importance of a common group identity for the individual decision-making and suggested it as a tool for improving cooperation and coordination in diverse teams. When people, apart from their own utility, value the group's or, in my context, the project's payoff, it increases the overall effort. Programmers in my sample reduce their contributions to or never join projects with a conflicting national identity. This effect seems to be stronger than the drop in collaboration due to poorer communication between team members, as the first strand of literature would suggest.

Third, by investigating the consequences of an interstate conflict on workplace behaviour, the project contributes to the literature on microeconomic effects of international or ethnic tensions (Fisman et al. 2013; Hjort 2014; Ksoll et al. 2010; Marx et al. 2015; Rohner et al. 2013a,b). The research in this area, apart from documenting the economic effects, has tried to identify the underlying mechanisms: the external political pressure, trust, national or ethnic preferences and social norms. The evidence so far has been mixed. Fisman et al. (2013) evaluate the consequences of Sino-Japanese political tensions and find a strong negative stock-market response for firms that depended on bilateral trade relations. Their evidence points to the large role of political pressure rather than that of potential consumer animosity in shaping the effect. Rohner et al. (2013a) argue for another channel: they show that a military conflict in Uganda enforced ethnic identity and decreased generalized trust toward "out-group" people, which in its turn inhibited inter-ethnic cooperation and slowed down the subsequent economic recovery. Hjort (2014) examines the effect of ethnic divisions on team productivity at a Kenyan flower firm. Using micro-level data and a convincing identification strategy, he argues that an external political conflict increased the taste for discrimination of a rival ethnic group, thus resulting in the misallocation of resources and lower productivity of heterogeneous teams.[1] The analysis performed in my paper complements the existing empirical work. Direct political pressure, though possible in other economic settings, could hardly impose restrictions on the open-source cooperation on GitHub. It is also unlikely that the conflict altered the beliefs of Ukrainian programmers regarding trustworthiness or quality of Russian colleagues. As in Hjort (2014), it seems that the shift in social preferences (the need to comply with the new identity prescription) can explain the drop in the Ukrainian-Russian collaboration.

The paper is organised as follows. Section 2 describes the setting: it outlines the conflict between Russia and Ukraine, discusses the particularities of online collaboration on GitHub, and presents the dataset. Section 3 presents the empirical strategy and discusses the effect of the conflict on overall collaboration between Ukrainian and Russian programmers. Section 4 presents the empirical strategy and discussess the effect of the conflict on project performance. Section 5 concludes.

2 The Setting

2.1 The Russian-Ukrainian Conflict

After the collapse of the Soviet Union, Ukraine had preserved close ties to Russia. In 2012, Russia was Ukraine's largest trading partner accommodating 25% of all Ukrainian exports and accounting for 32% of all Ukrainian imports.[2] As of the latest census in 2001, Russians, constituting 17% of Ukrainian population, represented the second largest ethnic group. Almost 30% of all Ukrainian citizens considered Russian their primary language.[3] Practically, however, the vast majority of the Ukrainian population was bilingual, and since the Ukrainian independence in 1991, no major ethnicity- or language-based conflict had ever occurred.

The internal crisis in Ukraine burst out in November 2013, when the former president Viktor Yanukovych suspended preparations for an association agreement with the European Union. This unilateral decision, which contradicted the previously promised policy,

[1] In recent work, however, Berge et al. (2016) conduct a series of "lab-in-the-field" experiments in Kenya and find no evidence for ethnic preferences. The authors call for more careful linking of the observed ethnic bias in behaviour and the actual ethnic bias in preferences.

[2] http://wits.worldbank.org/CountryProfile/en/Country/UKR/Year/2012/Summary.

[3] According to the latest available census data: http://2001.ukrcensus.gov.ua/eng/results/general/.

became the impetus for the series of so-called Euromaidan protests in November - February 2014. The protesters called for the resignation of Yanukovych and the government, with primary accusations being corruption and abuse of power. The protests culminated in February 2014 with intensive fights between protesters and police. In late February 2014, Yanukovych fled the country and a new government led by the opposition leaders came into office.

Meanwhile, anti-Euromaidan demonstrations started in Crimea and Eastern regions of Ukraine. There, the apparent impetus was given by the new parliament's vote (on the second day after Yanukovych fled) to abolish the Law on the State Language Policy. This law was adopted under the rule of Yanukovych and effectively gave Russian the status of the second state language in regions where at least 10% of the population reported Russian as their mother tongue. Although this decision never came in force, it easily became one of the headlines of the pro-Russian propaganda. In the end of February 2014, during clashes between pro- and anti-Euromaidan supporters in Crimea, troops without insignia (who turned out to be Russian military) silently took control over the Supreme Council of Crimea and the major military bases on the peninsula. After that, the Crimean parliament voted for a new government led by Sergey Aksyonov, the leader of the Russian Unity party. On March 16, 2014, the new Crimean government held the Referendum. According to the officially reported figures, the turnout reached 83.1%, and 96.77% of the voters supported the government's proposal: first, to declare independence from Ukraine and second, to become part of Russia.

The Referendum marked the beginning of the severe crisis between Ukraine and Russia. Shortly after the Crimean events, pro-Russian governments in Luhansk and Donetsk regions declared their independence from Ukraine. In April 2014, active military actions (Donbass War) between the separatists and the regular Ukrainian army began. In February 2015, after the peace negotiations in Minsk, both parties agreed to seize fire. Effectively, however, irregular fighting still takes place with both sides accusing each other of violating the peace agreements. As of mid 2016, the war had resulted in over 10,000 casualties, more than 22,000 wounded, and 1.4 million displaced persons.[4] Although the international institutions classify the war in the Eastern Ukraine as an "internal conflict", Russia is often accused of sending own troops and of supplying the separatists. In January 2015, the Ukrainian parliament voted on officially calling Russia the "aggressor country".

For both the Ukrainian and the Russian populations, the conflict came unexpectedly. Figure 1 illustrates changing attitudes of Ukrainians and Russians toward each other.

[4] http://www.ohchr.org/EN/NewsEvents/Pages/DisplayNews.aspx?NewsID=20496LangID=E

Figure 1: **The positive attitude of the population of Ukraine to Russia and of the population of Russia to Ukraine**

Notes: The graph shows the share (in percent) of the Ukrainian and the Russian populations who report positive or very positive attitudes toward the other country. Source: Kiev International Institute of Sociology

While in February 2014, about 80% Ukrainians still reported to have a positive attitude toward Russia, this number dropped to about 50% in the next poll conducted in May 2014, and reached the minimum of 30% in May 2015. The attitudes of Russians toward Ukrainians showed similar dynamics. While there have been some signs of recovery since December 2017, as of September 2019, the share of Ukrainians with a positive attitude toward Russia and vice versa remained below the pre-conflict values.

Unsurprisingly, the conflict had direct economic consequences. In 2015, Ukrainian exports to Russia dropped to constitute 12.1% of all Ukrainian exports, whereas the share of Russian imports declined to 20%.[5] Bilateral economic sanctions and increased tariff protection introduced and reinforced throughout 2014-2015 had directly contributed to the decline. Hence, for general economic collaboration, it is hard to separate the effects related to changes in people's attitudes (identities) from those imposed by political decisions. I try to overcome this problem by exploiting the setting where external factors, such as sanctions, should not play a role - the online platform for hosting software projects GitHub.

[5] http://stat.wto.org/CountryProfile/WSDBCountryPFView.aspx?Country=UA

2.2 GitHub

GitHub is a web-based repository hosting service, launched in 2007. It offers the infrastructure to store and to share software projects and provides several collaboration features such as code review, bug tracking, feature requests, task management, and wikis. As of May 2019, GitHub reports having more than 37 million users and more than 100 million repositories (projects), making it the largest host of source code in the world. Projects on GitHub can owned by individuals as well as by companies. GitHub users can choose between public and private repositories. GitHub provides the latter on the paid basis and allows restricting access to the general public. Project's founders can choose from a variety of licences to protect their code and to stipulate sharing rules.[6]

To begin working on GitHub, a person first registers as a user. During the registration, users choose their login and optionally report their name, location (in most cases, city), company, and biography. To start a project, a user creates a repository to store all source codes and related materials. Every project can have only one owner. The project owner can invite other GitHub users to collaborate, by offering them to become project members. Alternatively, an interested user can signal his willingness to become a project member by starting to contribute to the project (McDonald and Goggins 2013). Project members have the right to copy (in GitHub slang: fork) the code to their own repositories, directly modify the source code (in GitHub slang: commit) in the master repository, open and close issues related to the project. They can also remove themselves from the project without the agreement of the project owner. All other users, who are not project members, can copy the project's source code, report bugs and other issues, and *suggest* their own modifications (in GitHub slang: pull requests). The project owner or other project members then review these proposals and decide on whether to accept them or not. If a pull request is approved, the proposed modification is merged to the source code. Every GitHub user can observe who, when, and how much contributed to a project, provided the project is public. The profile page of every user displays projects he/she contributes to, together with the timeline of commits. Moreover, upon interest, users can directly examine written codes of each other. This feature makes GitHub a collaboration environment with low information asymmetry and low free-riding opportunities. Quality and quantity of contributions by each project member or external contributer can be directly observed. Apart from the collaboration environment, GitHub also offers some features of a social network. Users can get updates on the activities of other users they

[6] The available licences range from GNU General Puplic License, which literally make the source code open to anyone to the more restrictive Apache Licenses.

choose to "follow". Similarly to the "like" button on the Facebook, users can "star" an interesting project on GitHub or "watch" it to receive project updates.

Motivations of GitHub users differ.[7] My core dataset includes only public repositories, which mainly operate under open-source licenses. Economic research, starting with the works by Lerner and Tirole (2001, 2005) and followed, among others, by Belenzon and Schankerman (2008) and Hergueux and Jacquemet (2015) aimed at identifying motivations of open-source contributors. Programmers can be driven by pure economic incentives (software companies pay for working on their open-source projects or the platform is used to launch own new product), career concerns (contributing to an open-source project generates a positive signal about programming skills and increases programmers' visibility, among others, to the potential employers), utilitarian needs (developing software for own purposes), reciprocity, or pure altruism. Whereas, motivations of open-source contributors vary, the utility from their work relates to the success of the projects they are contributing to. The commonly accepted measures of projects' success are the number of commits, the number of committers, and the number of forks (Kalliamvakou et al. 2014; McDonald and Goggins 2013). Apart from that, GitHub combines a variety of activity indicators including stars, forks, commits, follows, and page views to construct ranks of projects and users. The ranks are updated in real-time and are visible to every one interested.[8]

2.3 Dataset

The main dataset is the GHTorrent database downloaded in October 2018.[9] GHTorrent collects all information from the GitHub public API and organises it in a relational database. The database records activities, which happen on the platform (public accounts), such as user registration, project creation, adding commits, issues, comments, etc. In the analysis, I use several tables from this database. The table "Users" contains all users ever registered on the GitHub. The table "Projects" records all repositories created at the GitHub. The tables "Commits" and "Issues" track activities related to projects and users, such as modifications of source code, adding comments, exchanging messages, bug tracking and fixing. I can also observe pull requests and their approval/decline, stars, and

[7] Substantial amount of GitHub users do not really collaborate, but rather use the platform for storage purposes. Consequently, many GitHub projects have only one committer. I exclude these cases from my analysis

[8] https://github.com/trending

[9] mysql-2018-10-01 downloaded from http://ghtorrent.org/downloads.html. Alternatively, it is possible to access several GHTorrent dumps through Google BigQuery services.

followers. Through the unique identifiers of users, projects, commits, and other events I can link the tables together. As the dataset is huge (over 300 GB), I make several restrictions to construct the working data sample. First, I consider only projects with at least two contributors and only those users who have at least one commit. Second, I select only those projects and events, which are related to users from the countries of interest (Ukraine, Russia, and several control countries, which will be discussed later). I determine the user nationality based on self-reported user locations and country codes provided by GitHub. Third, I aggregate all activity events (per user and project) by month. In order to control for overall activity levels of particular projects and users, I also calculate their total monthly activity levels (number of commits, issues, etc.). Table 3 in the Appendix summarizes the main data tables used in the analysis.

Most events in my dataset are automatically recorded by GitHub. Therefore, the measurement error related to the timing of different events (user registration, project creation, commits, etc.) is small. For each activity, GitHub generates a timestamp. Sometimes, dumps of huge databases such as GitHub suffer from "holes" in data (for example, due to a connectivity problem during the data dump, some observations may be missing). However, I do not expect this problem to affect activities of Ukrainian and Russian programmers differently from all other GitHub users. The main measurement error for my analysis can occur due to misreporting of user locations, which I use to identify the programmers' nationalities. I observe locations as of the data dump (October 2018). Part of the attenuation bias will come from users reporting wrong or non-existing locations. It is more problematic if some users adjusted their locations after the conflict (for example, Ukrainian programmers who still wanted to work with Russians without being "blamed" by others could have opened additional accounts or Russian users could have masked their real locations). Under such selective mis-reporting, I would get an upper-bound estimate of the effect's magnitude due to misclassifying users with the highest benefits from the collaboration. Yet, given the nature of GitHub, such misreporting is unlikely. First, reputation building is important for most users, and it accrues through the cumulated number of commits, followers, and stars, thus lowering incentives to hold multiple accounts. Second, complete activity history is usually observable, making it difficult for users to hide their information.

2.4 Descriptives

This section presents the descriptive statistics on activities of Ukrainian and Russian programmers on GitHub. Figures 7 and 8 in the Appendix illustrate that projects from

Ukraine and Russia follow the dynamics of other GitHub projects very closely. Figure 7 shows the monthly count of newly registered projects by region. I assign projects to regions: Ukraine, Russia, EU, or Overseas (US, Canada, Japan) based on the location of the project owner. Figure 8 presents the count of "star" events.[10] To generate these measures, I count how many stars projects from Ukraine, Russia, EU, and Overseas regions received per month. New projects create demand for potential contributions, while the amount of stars proxies projects' quality. There seems to be no Russia- or Ukraine-specific demand or quality shock during the analysed period. Hence, the inclusion of month fixed effects can absorb general changes in project activity level or their perceived quality.[11]

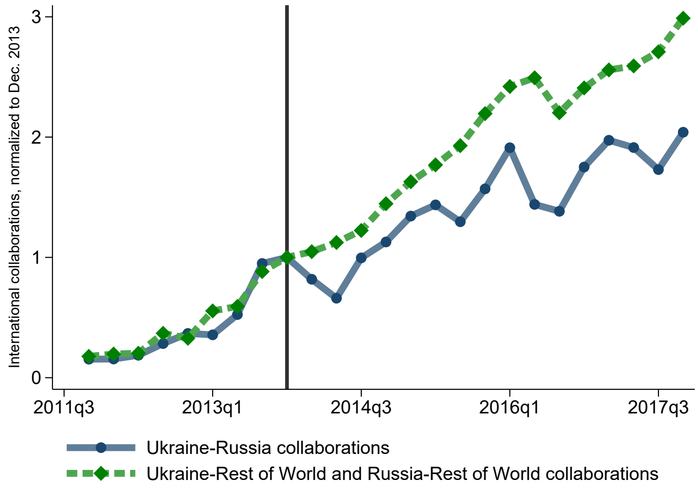
Figure 9 in the Appendix illustrates the activity level of Ukrainian and Russian programmers on the platform. I use the number of commits (source code modifications) as the main activity measure of GitHub users. Both Russian and Ukrainian users do not seem to be less active after February 2014: their number of commits increases almost every month. Figure 2 below refers to the international collaborations of Ukrainian and Russian users on GitHub. I consider a collaboration as international when a user from one country contributes to a project owned by a user from another country. The data are aggregated by quarter and normalised to quarter 4, 2013. The figure clearly shows that prior to the conflict Ukrainian-Russian collaborations developed in the same way as Ukrainian and Russian collaborations with other countries. Yet, in the first months after the conflict, the absolute number of Ukrainian-Russian collaborations dropped. For the next four years following the conflict, the growth rate of Ukrainian-Russian collaboration became substantially lower than that of their collaborations with other countries.

I will further validate this descriptive result with a regression analysis and will argue that enforcement of the national identity could have led to this drop in the collaboration between Ukrainian and Russian users.

[10] A "star" event is recorded whenever a user puts a star (a like) on a particular project.

[11] One may wonder about the observable drops in the time-series around May-December 2014 and July-December 2015. I am not aware of GitHub-specific changes in policy or technology, which could have provoked it. Most likely, these holes are due to technical problems during the data dump.

Figure 2: **International Collaborations of Russian and Ukrainian Programmers on GitHub**



Notes: An international collaboration is defined as a commit to a project from another country. The data are aggregated by quarter. The graph plots the normalized number of international collaborations (quarther 4, 2013 = 1). Blue line: the number of bilateral Russian-Ukrainian collaborations (Ukrainian user contributes to a Russian project and vice versa). Green line: the number of Russian collaborations with the rest of the world plus the number of Ukrainian collaborations with the rest of the world. The vertical black line corresponds to quarter 4, 2013.

3 Effect of the Conflict on the Overall Collaboration between Ukrainian and Russian Programmers

3.1 Empirical Approach

To estimate the effect of the conflict on the collaboration between Ukrainian and Russian programmers, I first focus on changes in the number of bilateral commits. A commit means any modification to the source code of a project. In comparison to other contributions (reporting issues, tracking bugs, adding comments), it is of the greatest value to any project and serves as the direct measure of the project's progress (McDonald and Goggins 2013). From a user perspective, commits represent their most visible activity and serve as the measure of their productivity. GitHub summarizes user commits by project directly on user profile pages (see Figure 11 in the Appendix).

To create the baseline sample, I select all commits done by Ukrainian and Russian programmers over 2012-17. I also add commits done by programmers from seven control countries: Belarus, Czechia, Hungary, Kazakhstan, Poland, Serbia, and Slovakia. Prior to the conflict, programmers from these countries had comparable activity levels and collaboration patterns as Ukrainian programmers. Through the unique identifiers I link commits to user and project data and aggregate them on month, user, and project level. From the user data, I can identify users' locations, company affiliations, entry date on the platform, followers if any. From the project data, I can identify a project's owner and assign a country to the project based on the reported country of the project's owner. In this way, a Ukrainian-Russian commit represents a commit when a Ukrainian user contributes to a project owned by a Russian user. In addition, for each project, I can observe the number of monthly commits by all other users, number and identifiers of project members, and received stars. I restrict the sample to include only those commits where user ID and project owner ID are different to focus on collaborative activities only, as the costs of contributing to own rather than to others' projects are very different and are not directly comparable. I thus drop more than 60% of all observations, which leaves me with about 1,000,000 user-project-month observations for programmers from Ukraine, Russia, and control countries. I further restrict the sample by dropping all commits and owned projects by programmers from Crimea, Luhansk, and Donetsk, as these regions could have been directly affected by the conflict.

For baseline regressions, I further aggregate the data on month, country of user, and region of project level.[12]. The goal of the empirical analysis is to capture how the conflict affected Ukrainian-Russian and Russian-Ukrainian collaborations relative to Ukrainian and Russian collaborations with other countries.

The baseline specification represents a triple difference and takes the following form:

$$Commits_{cpt} = \beta_0 + \beta_{11} * POST_t * Joint\_UA\_RU_{c,p} + \tag{1}$$

$$+\beta_{12} * POST_t * UA_cRU_p + \beta_2 * X_{pt^*} + \delta_{cp} + \gamma_{ct} + \phi_{pt} + \epsilon_{cpt}$$

where: c - country of contributor, p - region of project owner, t - month. $POST_t$ equals to one from February 2014. $Joint\_UA\_RU_{c,p}$ equals to one for Ukrainian-Russian or Russian-Ukrainian collaborations; it captures the relative change in the Ukrainian-Russian collaborations, without distinguishing between Ukrainian contributions to Russian projects or Russian contributions to Ukrainian projects. UA_cRU_p equals to one for collaborations when a Ukrainian user contributes to a Russian project; if it is different from zero, it means that Ukrainian users responded to the conflict differently from Russian users. X_{pt^*} denotes the total number of stars received by all projects from a given region over the last three months; it controls for time-varying quality of projects in a given

[12] Based on the location of a project's owner, I classify each project as Ukrainian, Russian, EU, Overseas, or Other

region. δ_{cp} are country of contributor*region of project fixed effects; they control for the average level of contributions in a given country-region pair. γ_{ct} and ϕ_{pt} are time-specific fixed effects respectively for the country of contributor and the region of project; they control for time-varying changes in the activity level of contributors from a given country and for time-varying changes in the demand for contributions from a given region.

Main dependent variable is the total number of commits per month in a given country-region pair. In addition, I use several measures to differentiate between intensive and extensive margins of collaboration. As an extensive measure, I use the number of unique users from a given country who contribute at least once to projects from a given region in a given month. I then use two measures to describe the intensive margine: 1) the number of different projects per user in a given month; 2) the amount of monthly contributions per user-project.

The difference-in-difference estimators β_{11} and β_{12} are consistent provided common pre-trends and the absence of time-varying unobservable factors, which independently from the confict could influence the collaboration patterns between Ukrainian and Russian programmers. While the specification 1 tries to control for the latter, to check for the absence of pre-trends, I estimate an event study. I center the data around quarter 4, 2013 and allow the coefficients corresponding to the Ukrainian-Russian and Russian-Ukrainian collaborations to vary across quarters before and after the conflict.

$$Commits_{cpt} = \beta_0 + \beta_{11} * Q_t * UA_c RU_p + \beta_{12} * Q_t * RU_c UA_p + \beta_2 * X_{pt*} + \delta_{cp} + \gamma_{ct} + \phi_{pt} + \epsilon_{cpt} \quad (2)$$

where: Q_t - event time dummies in quarters. β_{11} corresponds to Ukrainian-Russian collaborations (a Ukrainian user contributes to a Russian project) and β_{12} corresponds to Russian-Ukrainian collaborations (a Russian user contributes to a Ukrainian project). Other controls and fixed effects are the same as in the specification 1.

3.2 Results: Drop in the Collaboration

The regression analysis illustrates that the observable drop in the number of Ukrainian-Russian collaborations (Figure 2) was caused by the conflict rather than by some other Ukraine- or Russia-specific shocks.

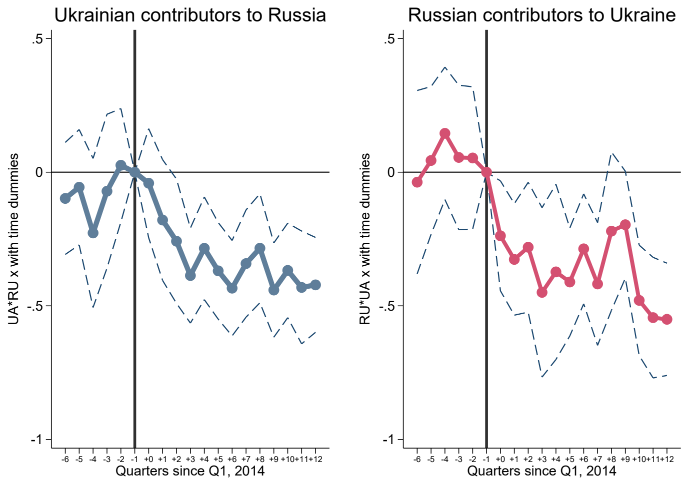
Table 1 reports the results of the baseline triple-difference estimations. Column one shows the effect on the total number of monthly collaborations. The coefficient for *JointUA*RU*Post* is negative and statistically significant, meaning that both Ukrainian and

Table 1: **The Effect of the Conflict on Ukrainian-Russian Collaborations on the GitHub**

VARIABLES	(1) Commits total	(2) Unique contributors	(3) Projects per user	(4) Contributions per user-project
Joint UA*RU*Post	-0.188**	-0.458***	0.0632**	0.192*
	(0.0775)	(0.0485)	(0.0299)	(0.105)
UA*RU*Post	-0.229*	0.103	-0.155***	-0.0562
	(0.134)	(0.0657)	(0.0528)	(0.159)
Stars, 3m	-0.0266	-0.241**	0.0724	0.00433
	(0.110)	(0.123)	(0.0716)	(0.189)
Observations	1,296	1,296	1,296	1,296
Pseudo R2	0.997	0.990	0.0373	0.127

Notes: All regressions control for Contributor country*Project region fixed effects, Month*Contributor country fixed effects, Month*Project region fixed effects. Estimation method: poisson. *Joint UA*RU*Post* denotes an interaction between Ukrainian-Russian and Russian-Ukrainian collaborations with a *Post* dummy. *UA*RU*Post* denotes an interaction between Ukrainian-Russian collaborations with a *Post* dummy; it checks for asymmetric responses to the conflict among Ukrainian and Russian programmers. Data spans 2012-17 years and comprises all contributions by programmers from Ukraine, Russia, and control countries.

Figure 3: **Ukrainian-Russian Collaboration: Quarterly Treatment Effect of the Conflict**



Notes: The graphs plot the interaction coefficients between quarterly time dummies and the indicators for Ukrainian-Russian (left panel) and Russian-Ukrainian (right panel) collaborations. I use the extensive margin measure: the number of unique users from a given country contributing to projects from a given region. Controls: Contributor country*Project region fixed effects, Month*Contributor country fixed effects, Month*Project region fixed effects. Estimation method: poisson. The vertical black line corresponds to the reference period (Q4 2013).

Russian programmers reduced their monthly contributions to each other's projects (relative to their contributions to projects from other countries) following the conflict. The coefficient for *UA*RU*Post* is also negative and significant, meaning that Ukrainian programmers reacted more strongly: they reduced their monthly contributions to Russian projects by 0.23 log points more compared to the average drop in Russian contributions to Ukrainian projects.

Column two shows the effect on the extensive margin as measured by the number of unique contributors to projects from a given region in a given month. The coefficient for *JointUA*RU*Post* is again negative and statistically significant. It is also larger in magnitude compared to the total effect. This means that the conflict mainly affected the extensive margin of collaboration: it reduced the number of Ukrainian/Russian users contributing to Russian/Ukrainian projects in a given month by 0.46 log points (again relative to collaborations with other countries). The coefficient $UA * RU * Post$ is not statistically significant, meaning that the responses of Ukrainian and Russian programmers were similar on the extensive margin. Figure 3 confirms that there were no pre-trends prior to the conflict. It also shows that three years following the conflict, the negative effects persist.

Column three illustrates the effect on the intensive margin as measured by the number of different projects per contributing user in a given month. Here, I observe differences in the responses between Russian and Ukrainian programmers. Conditional on contributing to Ukrainian projects after the conflict, Russian programmers slightly increase the number of projects they work on. In contrast, Ukrainian programmers reduce the amount of Russian projects they work on. This asymmetric decrease on the intensive margin also explains why the total effect for Ukrainian-Russian collaborations is stronger than that for Russian-Ukrainian collaborations.

Finally, column four presents the effect on the intensive margin as measured by the number of monthly contributions per given user working on a given project. The results show that conditional on contributing to a project, neither Russian, nor Ukrainian programmers drop the amount of their monthly contributions.

3.3 Discussion of the Mechanisms

The regression results confirm that the conflict had a negative effect on the Ukrainian-Russian collaboration, *relative* to the Ukrainian and Russian collaborations with other countries.[13] This relative drop in collaboration was mainly driven by the extensive margin: fewer Ukrainian and Russian users contributed to each other's projects after the conflict. In addition, on the intensive margin Ukrainian users reduced the amount of Russian projects they work on.

My specification ensures that the results are not driven simply by lower activity of Ukrainian and Russian programmers on the platform or by lower demand for contributions by Ukrainian or Russian projects. In the following subsection, I first investigate whether the observed effects could be still driven by economic considerations, such as career concerns, or whether they could be attributed to higher social, identity-driven, costs of collaboration.

3.3.1 Changing Career Considerations

It might be that the main motivation to commit to a particular project is to increase own visibility in front of a potential employer. If Russian companies lose their attractiveness due to the economic crisis that burst out in 2014, fewer Ukrainian programmers would seek a job in Russia. The incentives to contribute to Russian projects would diminish, leading to a smaller number of commits. This could still represent the indirect effect of the conflict (introduced sanctions against Russia account for some of the economic decrease), however, it is not related to the identity. The triple-difference estimations, with users from other countries as the control group, can partly alleviate these concerns. For instance, career opportunities in Russia for Belarusian or Kazakh programmers due to language similarity and the intensity of previous relations should be very similar to those of their Ukrainian colleagues. Therefore, if the bad economic situation makes Russian companies and projects less attractive, it should also influence the flow of contributions from the control countries. The results presented earlier in Table 1 and Figure 3 however, do not confirm this explanation, as the interaction term $JointUA * RU * Post$ is significant conditional on the time-specific project region fixed effects, which net out the general economic decline in Russia following the conflict.

Another possibility is that Ukrainian programmers, who used to work on GitHub for Russian companies, were affected by real or anticipated increase in bilateral transaction costs. Higher uncertainty about future collaboration costs (for example, due to possible interruptions in banking services between Russia and Ukraine) could make Ukrainian pro-

[13] In this way, the coefficients should not be fully interpreted as an absolute drop in collaboration: first, the overall activity of Ukrainian and Russian programmers on the platform increases; second, as they reduced their contributions to each other's projects, they could have freed some time to increase their contributions to other regions.

grammers withdraw from Russian projects. To alleviate this concern, I compare changes in Ukrainian-Russian commits to projects owned by companies vs. those owned by individual users. If bilateral transaction costs mattered, I would expect a stronger decrease in contributions to companies. I reestimate quarterly treatment effects separately for contributions directed at companies and for contributions directed at individual users. Figure 12 displays the results, which point in the opposite direction: Ukrainian-Russian contributions to projects owned by companies (rather than individuals) are not affected by the conflict.

3.3.2 Identity-driven Concerns

To provide more support for the identity effect, I analyse differences among Ukrainian programmers, for whom the role of external factors should be the same, but the identity effects could be different. One dimension to vary is the strength of the national identity. The larger the importance of the national identity for an individual, the more he or she is likely to react to an international conflict, which increses the salience of the national identification.[14] While the true national identity cannot be observed, I exploit ethnic heterogeneity within Ukraine to construct a proxy for it.

I assign a Ukrainian programmer a stronger national identity if he or she comes from a district with fewer ethnic Russians. To that end I use data from the 2001 census that reports on ethnic and language composition of Ukrainian districts.

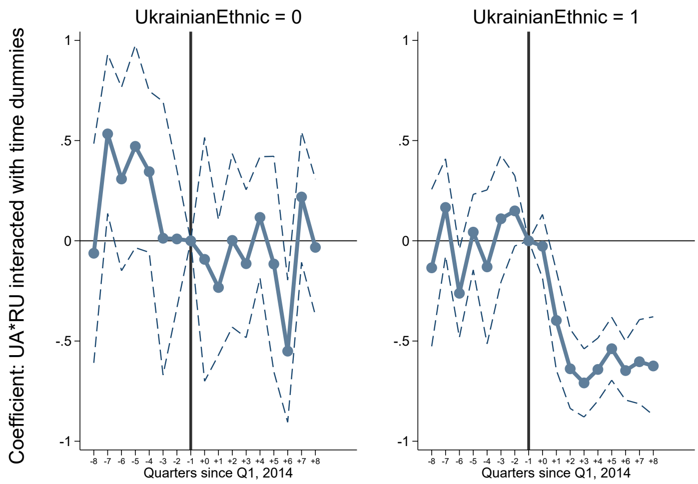
Figure 4 separately estimates the change in the Ukrainian-Russian collaborations. Left panel keeps only those Ukrainian programmers who come from districts with relatively more ethnic Russians (the share of ethnic Russians is above the median for Ukraine), while right panel keeps only those Ukrainian programmers who come from districts with relatively few ethnic Russians. While the conflict does not seem to affect Ukrainian-Russian collaborations in the former case, it significantly reduces the collaboration in the latter case.

3.3.3 Effect of the Conflict on Team Formation

I further decompose the effect of the conflict by distinguishing between direct and indirect contributions. On GitHub, both project members and external users can contribute to a project. Respectively, I denote as 'direct' contributions by project (team) members and 'indirect' - those by other GitHub users. The main difference is that project members

[14] For instance, in a recent experimental study Mechtel et al. (2016) show that the revealed strength of identification predicts subsequent allocation choices.

Figure 4: **Ukrainian-Russian Collaboration: Quarterly Treatment Effect of the Conflict by Ethnic Composition**



Notes: The graphs plot the interaction coefficients between quarterly time dummies and the indicators for Ukrainian-Russian collaborations when a Ukrainian programmer comes from a city with the share of ethnic Russians > median (left panel) and when a Ukrainian programmer comes from a city with the share of ethnic Russians <= median (right panel). I use the extensive margin measure: the number of unique users from a given country contributing to projects from a given region. Controls: Contributor country*Project region fixed effects, Month*Contributor country fixed effects, Month*Project region fixed effects. Estimation method: poisson. The vertical black line corresponds to the reference period (Q4 2013).

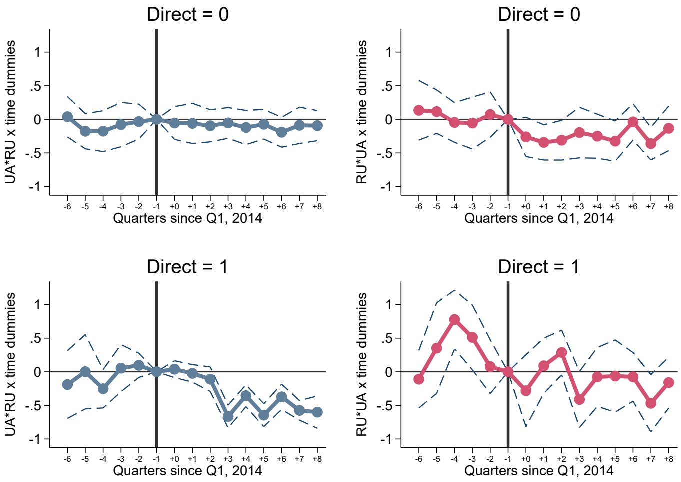
have the right to *directly* modify the project by adding their commits, whereas the modifications by external contributors will be merged only after a project owner or a project member approves them.[15]

My baseline estimates pooled direct and indirect contributions. I now decompose the effects by distinguishing between the two. There are three reasons for such decomposition. First, compared to indirect commits, direct contributions (by a project member) usually require more committment and closer informal coordination with other project members to avoid conflicts. Second, direct contributions appear more saliently on a programmer's profile and hence are more visible to other users. Third, the decision whether a direct contribution is merged in the project depends on the contributor him/herself, whereas an indirect contribution may be rejected by another programmer.

Figure 5 separately estimates the quarterly treatment effects for direct and indirect contributions. It displays interesting heterogeneity among Ukrainian and Russian programmers. Ukranian users significantly reduce their direct contributions to Russian

[15]The technical difference between direct and indirect contributions is that direct ones appear in the master project, whereas indirect commits must be linked to the local copies (forks) of the master project (as the external users do not have access rights to directly modify the source code). Because I can observe the "parent" for each forked project, I can link all the commits to the actual master projects.

Figure 5: **Ukrainian-Russian Collaboration: Quarterly Treatment Effect of the Conflict, Direct vs. Indirect Contributions**



Notes: The graphs plot the interaction coefficients between quarterly time dummies and the indicators for Ukrainian-Russian external collaborations (left upper panel) and internal collaborations (left lower panel). The right panel illustrates corresponding interaction coefficients for Russian-Ukrainian external and internal collaborations. I use the extensive margin measure: the number of unique users from a given country contributing to projects from a given region. Controls: Contributor country*Project region fixed effects, Month*Contributor country fixed effects, Month*Project region fixed effects. Estimation method: poisson. The vertical black line corresponds to the reference period (Q4 2013).

projects, while no significant effect is detected for indirect ones. It is mainly driven by the extensive margin: following the conflict, Ukrainian programmers are less likely to join Russian projects as team members. Ukrainian programmers might turn away from direct and salient collaboration with Russians, because their preferences for teammates have changed following the conflict or because they do not want to be considered by other users as collaborating with Russians. Disentangling these two reasons remains an interesting avenue for further research. The response of Russian users is asymmetric. While direct contributions to Ukrainian projects are not affected, there is a significant negative effect for indirect contributions. This could be consistent with the fact that Ukrainian project owners become less likely to accept a contribution from a Russian user following the conflict.

4 Effect of the Conflict on Project Performance

Did the change in collaboration result in real productivity effects? In this section, I evaluate the consequences of the conflict for project performance.

4.1 Empirical Approach

The difference-in-difference approach compares the performance of the affected projects before and after the conflict relative to the control group. As affected (treated), I consider projects that received both Russian and Ukrainian commits before March 2014. I start with a sample of GitHub projects that received at least one commit by users from Ukraine, Russia, and the above control countries in 2012-2015. I further restrict the sample: first, I drop all projects with only one committer (to focus on collaborative work only) and, second, I consider only project that were active throughout February 2013-February 2014. This leaves a sample of 30,808 unique projects. I then assign treatment status if at least one Russian and one Ukrainian programmer collaborated on a project within the same month.[16] With the disaggregated data, I can identify all the projects with Russian and Ukrainian collaborations, without restricting the nationality of project owners. In the treated sample, owners from EU, Russia, and Ukraine account each for about 10% of projects, Overseas users own 18% of the projects, and about 50% of owners are located in other countries.

As Table 4 in the Appendix illustrates, the treated projects are very different from an average GitHub project: on average, they are twice as old, have more project members, and over February 2013-February 2014 they received several times more commits and stars. This is due to the fact that despite seemingly low collaboration costs, the majority of GitHub projects, similar to scientific research or patent production, feature strong localisation bias. Only projects of the highest quality manage to attract committers from a different location or company. Therefore, average local GitHub projects cannot constitute a reasonable control for the treatment group. In order to identify comparable projects, I apply the coarsened exact matching procedure (see Blackwell et al. (2009)) to match projects on age, programming language, region of owner (Russia, Ukraine, EU, Overseas, Other), number of project members, and number of commits over February 2013-February 2014. I manage to match 690 treated projects (from 272 bins). Columns 3 and 4 in the Table 4 illustrate that the pre-conflict descriptives for the matched sample are now well balanced.

[16] Alternatively, I define the continuous treatment by calculating the shares of Ukrainian and Russian commits to total project commits within one year preceding the conflict and construct the treatment variable as the $min\{share_{ua}, share_{ru}\}$.

I estimate the following difference-in-difference regression:

$$\begin{aligned}Y_{jt} = \beta_0 + \beta_{11} * RU * TREAT_j * POST_t + \beta_{12} * UA * TREAT_j * POST_t +\\ + \beta_{13} * TREAT_j * POST_t + \delta_{pt} + TREAT_j * \kappa_p + \beta_2 * X_{jt} + \epsilon_{jt}\end{aligned} \tag{3}$$

In Specification (3), Y_{jt} represents one of a project's j monthly performance measures: total number of commits (project's progress) or number of stars (project's popularity), both in natural logarithms. The main coefficients of interest are β_{11}, β_{12}, and β_{13}. β_{13} measures whether performance of all treated projects changed relative to the matched control group after the conflict. For example, if some users decided to leave their projects to not work together with people from a "hostile" country or if coordination costs within a Russian-Ukrainian team increased, this could have negatively impacted the flow of commits and, consequently, the project's popularity. β_{11} and β_{12} allow for a different treatment effect for projects owned by Russians or by Ukrainians. δ_{pt} are project-region by month fixed effects. As previously, region p is defined by the location of the project's owner. $TREAT_j * \kappa_p$ is the interaction between the treatment indicator and the owner's region fixed effect to account for possible region-specific differences of treated projects at the baseline. X_{jt} are project-specific controls, such as owner type (company or individual) or the pre-conflict number of commits. Provided the matching of the treatment and control projects was successful, the inclusion of terms $TREAT_j * \kappa_p$, and $\beta_2 * X_{jt}$ should not affect the coefficients of interest.

4.2 Results: Project Performance

Table 2 reports the estimation results. In all the specifications I use monthly project-level data from 690 "treated" projects and 690 matched "control" projects identified as described above. The control group comprises similar projects, to which either Ukrainian or Russian programmers contribute, but which did not have a Russian-Ukrainian collaboration within one year preceding the conflict. The dummy $TREAT_jPOST_t$ shows the difference between treated and untreated projects after the conflict. In addition, I allow for the differential treatment effects for projects owned by Russians and by Ukrainians. For the third (neutral) countries' projects, the conflict's effect should come mainly through additional communication costs and lower coordination between Russian and Ukrainian team members. For projects owned by either Russian or Ukrainian users, the conflict effect has an additional channel: it imposes the stigma costs of helping the "enemy". Moreover, GitHub displays all open projects, to which a user contributes, on the profile page, thus amplifying the public image concerns. All specifications include month by project-region fixed effect, treat by project-region fixed effect, and programming language fixed effect. I conservatively cluster standard errors by project-region and programming language.

Table 2: **The Effect of the Conflict on Project Performance**

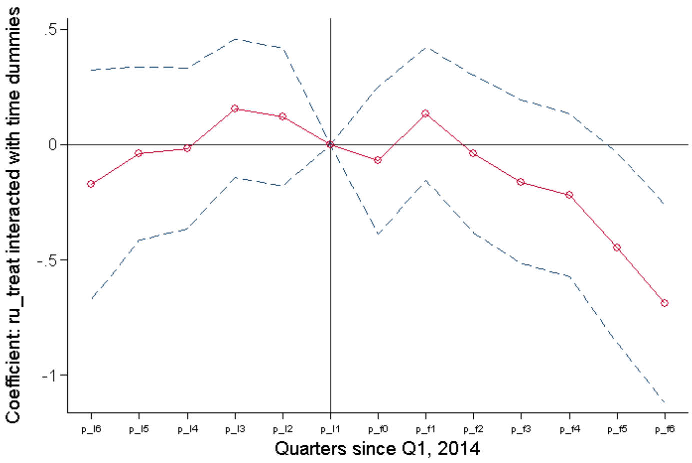
	(1) Total commits	(2) Commits by UA	(3) Commits by RU	(4) Stars
$RU\_TREAT_j POST_t$	-0.270	-0.173***	0.219	-0.0869*
	(0.189)	(0.0589)	(0.190)	(0.0465)
$UA\_TREAT_j POST_t$	0.140	0.253	-0.0124	-0.112
	(0.384)	(0.220)	(0.181)	(0.175)
$TREAT_j POST_t$	0.0271	-0.0798**	-0.0830	0.0477
	(0.0950)	(0.0370)	(0.0537)	(0.0365)
Pre-confl. commits	0.701***	0.0584***	0.151***	0.0585***
	(0.0231)	(0.0111)	(0.0179)	(0.00984)
Company	-0.206***	0.0256	-0.0736	0.00903
	(0.0661)	(0.0372)	(0.0531)	(0.0534)
Observations	31,954	31,954	31,954	31,954
R^2	0.544	0.168	0.188	0.080
Treat*project-region FE	yes	yes	yes	yes
Month*project-region FE	yes	yes	yes	yes
Language FE	yes	yes	yes	yes
Robust	yes	yes	yes	yes
Clusters	83	83	83	83

Notes: The dependent variable are: Columns 1 *ln*(*total commits*+1); Columns 2 and 3 *ln*(*commits by* UA *or RU users*)+1; Column 4 *ln(stars+1). All variables are measured per month* $RU(UA)$ $TREAT_j POST$ - a post period dummy for treated projects owned by Russian or Ukrainian programmers; $TREAT_j POST$ - a post period dummy for all treated projects. $Pre - confl.\ commits$ control for the total number of commits to a project before March 2014. *Company* - a dummy for projects owned by companies. All specifications include $month * project - region$, $treat * project - region$, and programming language fixed effects.

Column 1 displays the effect for the total number of commits to a project. Total number of commits captures the progress of a project and can be treated as a proxy of success. While there seems to be in general no difference between treated and untreated projects, the treatment effect for projects owned by Russians is negative. While in the presented Specification the coefficient is not statistically significant (p-value = 0.153), the coefficient gains significance with less conservative standard errors. Figure 6 shows the absence of pre-trends and reveals that the negative effect for Russian projects becomes sizable in about a year after the conflict started. Column 2 shows that, consistently with the results in the previous section, Ukrainian programmers significantly stronger reduce their commits to the treated Russian projects. Although, in response, the amount of contributions from Russian users increases, as column 3 reports, it does not fully compensate for the reduction.

Another observation, is that both Ukrainian and Russian users commit less to the treated projects from the third countries. This effect is not strong, but still indicates the presence of possible communication problems in mixed teams following the conflict. Column 4 compares treated and control projects in terms of their popularity, as measured by the amount of new stars, which a project receives from other users. As with the total performance, the coefficient is negative only for the treated Russian projects. This effect is estimated *beyond* possible general negative effects, which might arise for all Russian projects after the start of the conflict. It can be, therefore, attributed to the loss in value of the treated projects due to lower commits from Ukrainian team members.

Figure 6: **Total Commits to the Affected Russian Projects: Quarterly Treatment Effect of the Conflict, Triple Difference**



Notes: The graph plots the interaction coefficients between quarterly time dummies and the indicator for treated projects owned by Russian users. Dependent variable: ln(total commits) aggregated by month and project-region. The vertical black line corresponds to the reference period (Q4 2013).

5 Conclusion

This project studies the role that the national identity plays for the collaboration in existing and formation of new diverse teams. To conduct the analysis, I use data from GitHub - the world's largest hosting platform for software projects. For identification, I exploit the sharp aggravation in Russian-Ukrainian political relations due to the unexpected annexation of Crimea in March 2014. Following the event, collaboration between Ukrainian and Russian programmers fell significantly relative to their collaboration with other countries. This decrease cannot be explained by infrastructure constraints, nor by general lower activity of Ukrainian and Russian programmers, nor by economic reasons such as higher bilateral transaction costs or changing career concerns. I provide additional evidence for the role of identity in shaping this effect: Ukrainian programmers identified as having a stronger national identity react stronger to the conflict relative to other Ukrainian programmers.

This project has policy implications for firms managing or planning to attract diverse teams. The empirical evidence emphasises the risks of diverse teams due to their exposure to external factors. My results further show that the identity conflict not only hinders peer interaction within a diverse team, but also changes the preferences of some team members towards projects that are associated with a "hostile" social group. Therefore, in case of an identity-based conflict, having a third "neutral" party to lead team's work or enforcing a common project identity might be beneficial.

My results can be also relevant for open-source platforms, such as GitHub, that aim at facilitating international "barrier-free" collaboration. One option would be to hide the country of origin information before the real value of the collaboration is revealed.

Lastly, this project could be interesting for the general public by bringing to awareness the role of the national identity. Usually, people blame politicians and external factors for creating constraints to their collaboration. However, my results show that even in a setting with negligible legal or physical barriers, educated and informed people, still choose to follow the identity prescriptions at the cost of economically beneficial collaboration.

References

Akerlof, George A. and Rachel E. Kranton (2000). "Economics and Identity". *The Quarterly Journal of Economics* 115 (3), pp. 715–753.

Bandiera, Oriana, Iwan Barankay, and Imran Rasul (2005). "Social Preferences and the Response to Incentives: Evidence from Personnel Data". *The Quarterly Journal of Economics* 120 (3), pp. 917–962.

— (2009). "Social connections and incentives in the workplace: Evidence from personnel data". *Econometrica* 77 (4), pp. 1047–1094.

Belenzon, Sharon and Mark Schankerman (Oct. 2008). *Motivation and Sorting in Open Source Software Innovation*. CEPR Discussion Papers 7012. C.E.P.R. Discussion Papers.

Berge, Lars Ivar Oppedal et al. (Nov. 2016). *How Strong are Ethnic Preferences?* NBER Working Papers 21715. National Bureau of Economic Research, Inc.

Blackwell, Matthew et al. (2009). "cem: Coarsened exact matching in Stata". *Stata Journal* 9 (4), pp. 524–546.

Charness, Gary, Luca Rigotti, and Aldo Rustichini (2007). "Individual Behavior and Group Membership". *American Economic Review* 97 (4), pp. 1340–1352.

Chen, Roy et al. (2014). *Does team competition increase pro-social lending? Evidence from online microfinance*. Discussion Papers, Research Unit: Market Behavior SP II 2014-209. Social Science Research Center Berlin (WZB).

Chen, Yan and Sherry Xin Li (2009). "Group identity and social preferences". *The American Economic Review*, pp. 431–457.

Duflo, Esther (2001). "Schooling and Labor Market Consequences of School Construction in Indonesia: Evidence from an Unusual Policy Experiment". *American Economic Review* 91 (4), pp. 795–813.

Fisman, Raymond, Yasushi Hamao, and Yongxiang Wang (2013). "The impact of interstate tensions on economic exchange: Evidence from shocks to Sino-Japanese relations". *Unpublished working paper. Columbia University.*

Hergueux, Jme and Nicolas Jacquemet (2015). "Social preferences in the online laboratory: a randomized experiment". *Experimental Economics* 18 (2), pp. 251–283.

Hjort, Jonas (2014). "Ethnic divisions and production in firms". *The Quarterly Journal of Economics* 129 (4), pp. 1899–1946.

Kalliamvakou, Eirini, Georgios Gousios, and Blincoe Kelly (2014). "The promises and perils of mining GitHub (Extended Version)". *MSR 2014 Proceedings of the 11th Working Conference on Mining Software Repositories* (1), pp. 92–101.

Ksoll, Christopher, Rocco Macchiavello, and Ameet Morjaria (2010). "DP8074 The Effect of Ethnic Violence on an Export-Oriented Industry".

Lazear, Edward P (1999). "Globalisation and the Market for Team-Mates". *Economic Journal* 109 (454), pp. C15–40.

Lerner, Josh and Jean Tirole (2001). "The open source movement: Key research questions". *European Economic Review* 45 (4-6), pp. 819–826.

— (2005). "The Economics of Technology Sharing: Open Source and Beyond". *Journal of Economic Perspectives* 19 (2), pp. 99–120.

Marx, Benjamin, Vincent Pons, and Tavneet Suri (2015). "Diversity and team performance in a kenyan organization".

Mas, Alexandre and Enrico Moretti (2009). "Peers at Work". *American Economic Review*, pp. 1996–2003.

McDonald, Nora and Sean Goggins (2013). "Performance and Participation in Open Source Software on GitHub". *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. CHI EA '13. Paris, France: ACM, pp. 139–144. ISBN: 978-1-4503-1952-2. DOI: 10.1145/2468356.2468382.

Mechtel, Mario, Florian Hett, and Markus Krll (2016). *Endogenous Social Identity and Group Choice*. Annual Conference 2014 (Hamburg): Evidence-based Economic Policy 100307. Verein fr Socialpolitik / German Economic Association.

Moser, Petra and Alessandra Voena (2012). "Compulsory Licensing: Evidence from the Trading with the Enemy Act". *American Economic Review* 102 (1), pp. 396–427.

Prat, Andrea (2002). "Should a team be homogeneous?" *European Economic Review* 46 (7), pp. 1187–1207.

Rohner, Dominic, Mathias Thoenig, and Fabrizio Zilibotti (2013a). "Seeds of distrust: conflict in Uganda". *Journal of Economic Growth* 18 (3), pp. 217–252.

— (2013b). "War signals: A theory of trade, trust, and conflict". *The Review of Economic Studies* 80 (3), pp. 1114–1147.

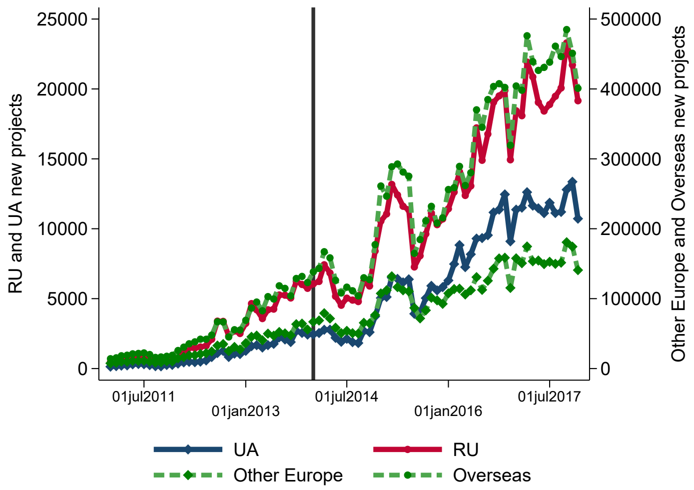
6 Additional Tables and Graphs

Table 3: **Summary of Github Tables**

GitHub Table	Variables
Users	user id, registration date, deleted (dummy), fake (dummy), login, name, email, country code (generated), location (self-reported), company name, type (organisation or individual)
Projects	project id, registration date, deleted (dummy), url, owner id, name, description, language, forked from
Events (commits, issues, pull requests, stars, follows)	date, event type, user id, project id

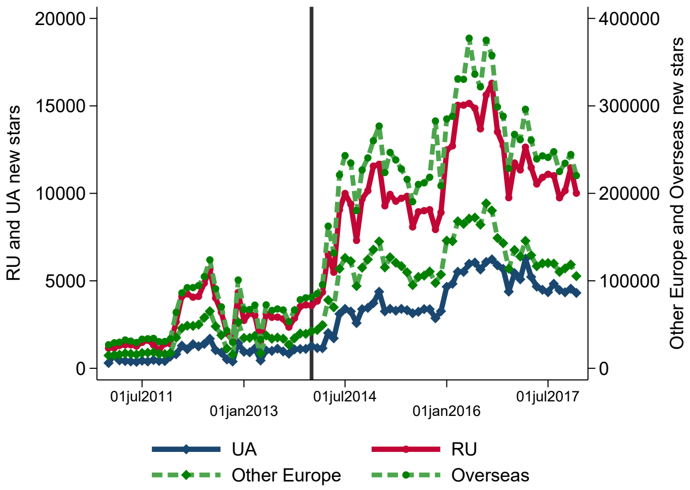
Notes: GitHub Data Dump from http://ghtorrent.org/

Figure 7: **Registration of New Projects on GitHub**



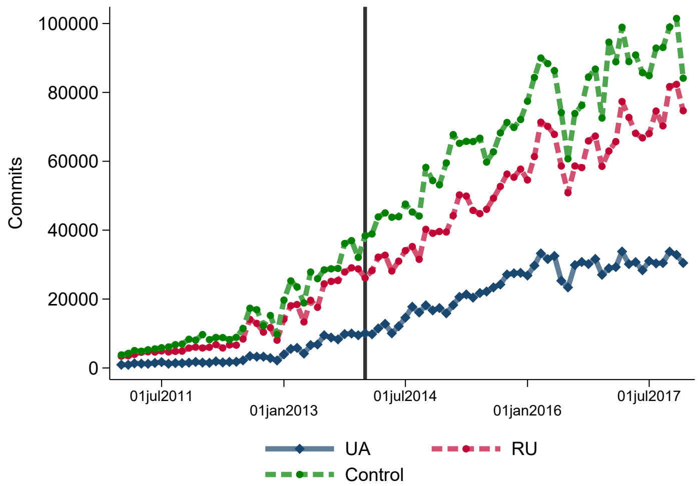
Notes: The graph plots the amount of newly registered projects on GitHub by month. The region is determined by the location of the project owner. To accommodate different activity levels, the left y-axis accounts for Ukrainian and Russian projects, the right y-axis - for EU and Overseas (US, Canada, Japan) projects. The vertical black line corresponds to February 2014.

Figure 8: **New "Star Events" on GitHub**



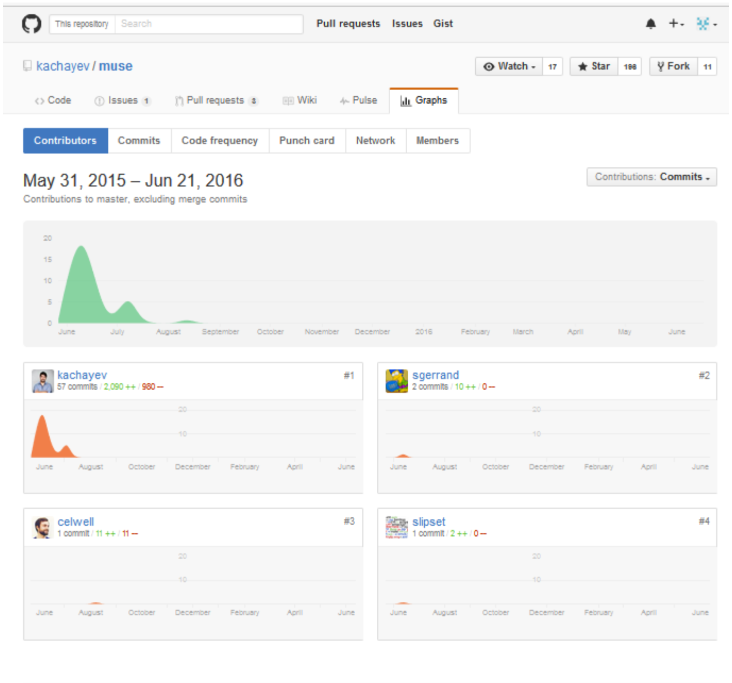
Notes: A "star" event is recorded whenever a user puts a star (a like) on a particular project. The amount of stars is used by GitHub as one of the quality measures. The region is determined by the location of the owner, whose project is "starred". The vertical black line corresponds to February 2014.

Figure 9: **Commits by Ukrainian, Russian and Control-group Users on GitHub**



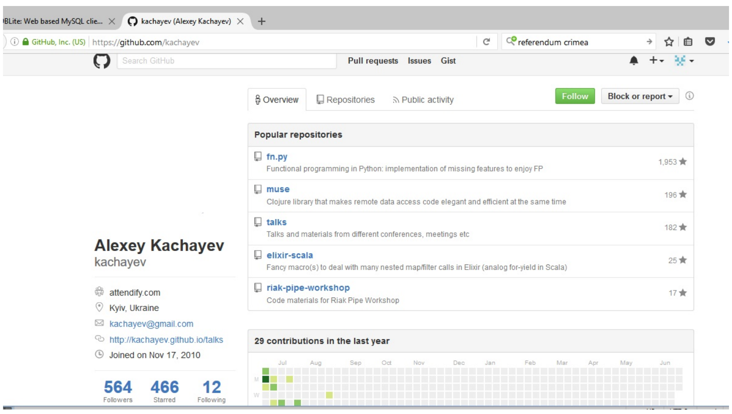
Notes: A "commit" event is recorded whenever a user modifies the source code of a project. The graph shows commits by programmers from Ukraine, Russia, and Control countries (Belarus, Czechia, Hungary, Kazakhstan, Poland, Serbia, Slovakia) to projects owned by other GitHub users. The vertical black line corresponds to February 2014.

Figure 10: **Profile of a Public Github Project**



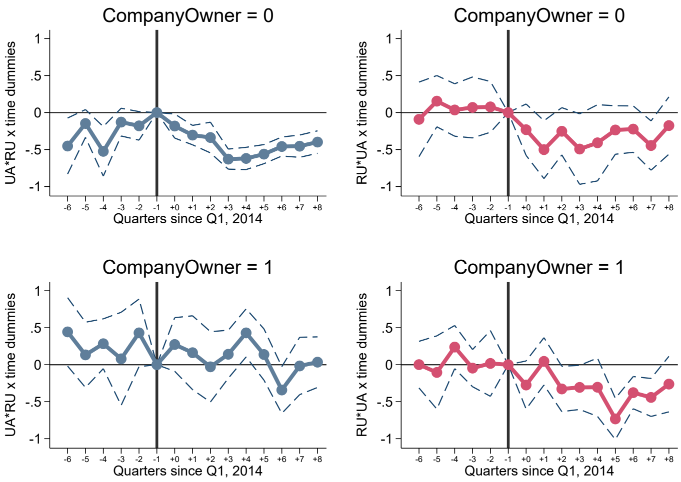
Notes: This view of a project is accessible to all registered Github users.

Figure 11: **Profile of a Github User**



Notes: This view of a user profile is accessible to all registered Github users. Github links the user profile to all public projects, to which a user has contributed.

Figure 12: **Ukrainian-Russian Collaboration: Quarterly Treatment Effect of the Conflict, Company vs. Individual Projects**



Notes: The graphs plot the interaction coefficients between quarterly time dummies and the indicators for Ukrainian-Russian collaborations when a project owner is an individual (left upper panel) and collaborations when a project owner is affiliated to a company (left lower panel). The right panel illustrates corresponding interaction coefficients for Russian-Ukrainian external and internal collaborations. I use the extensive margin measure: the number of unique users from a given country contributing to projects from a given region. Controls: Contributor country*Project region fixed effects, Month*Contributor country fixed effects, Month*Project region fixed effects. Estimation method: poisson. The vertical black line corresponds to the reference period (Q4 2013).

Table 4: **Treated and Control Projects on GitHub**

	(1) treat = 0 all	(2) treat = 1 all	(3) treat = 0 matched	(4) treat = 1 matched	(5) Diff. T-C (se)
Continuous treat	7.83e-05 [0.00450]	0.0426 [0.100]	2.09e-05 [0.000424]	0.0434 [0.101]	0.04*** (0.00)
Commits	83.33 [529.5]	540.6 [1,802]	446.0 [2,187]	483.0 [1,304]	37.00 (96.92)
Commits, weight.	35.71 [247.5]	248.0 [824.7]	216.5 [1,073]	223.5 [614.3]	7.04 (47.07)
Mean commits	16.15 [62.22]	59.65 [180.6]	55.48 [244.3]	54.19 [133.2]	-1.29 (10.59)
Commits, w/t RU and UA	74.93 [522.8]	476.4 [1,639]	428.1 [2,179]	426.2 [1,261]	-1.92 (95.83)
Number of members	0.282 [1.935]	0.637 [4.816]	0.351 [1.494]	0.457 [1.685]	0.11 (0.09)
Project age, m.	12.75 [14.01]	19.15 [16.94]	18.67 [16.51]	18.75 [16.65]	0.08 (0.89)
Stars	1.464 [15.84]	4.733 [25.95]	3.622 [19.63]	4.725 [26.16]	1.10 (1.25)
Stars, w/t RU and UA	1.456 [15.75]	4.703 [25.81]	3.591 [19.48]	4.694 [26.01]	1.10 (1.24)
Observations	30,105	703	690	690	1,380

Notes: The table compares projects with mixed Russian-Ukrainian teams (treated) to other projects on GitHub. Columns 1 and 2 compare all treated and non-treated projects. Columns 3 and 4 compared treated and non-treated projects in the matched sample. Column 5 shows the differences in values between Columns 4 and 3.